WHAT IS CLAIMED IS:

1. A method for performing timing analysis on a user design that has been placed on a programmable integrated circuit, the method comprising:

generating edge masks to annotate edges in a graph that represents at least a portion of the user design, the edge masks indicating whether a source or destination point is reachable from a corresponding one of the edges; and

performing at least one depth first search along a time critical path in the graph between the source and the destination points, the at least one depth first search being prevented by the edge masks from analyzing paths that do not lead to the source point or the destination point.

2. The method of claim 1 further comprising:

calculating slack and slack ratio values for edges in the time critical paths; and

if delay along any of the time critical paths exceeds the user timing constraint, modifying placement of the user design within the programmable integrated circuit using the slack and the slack ratio values.

3. The method of claim 2 wherein generating the edges masks further comprises generating a first type of edge mask that represents reachability from the source points in a backwards direction, and a second type of edge mask that represents reachability to the destination points in a forward direction.

4. The method of claim 3 wherein generating the edge masks further comprises generating binary bits that correspond to specific types of the source points and the destination points in a list of constraints.

5. The method of claim 4 wherein performing the at least one depth first search further comprises performing multiple depths first searches in the presence of multiple constraints on timing analysis, wherein source edge-masks correspond to source-types in timing constraints and source types correspond to source edge-masks.

6. The method of claim 5 further comprising:

converting multicycle constraints on timing analysis to two edge-mask sets, one for a base case and another for a multicycle case.

7. The method of claim 5 further comprising:

converting thru-x constraints into multiple edge-mask sets indicating a base case and a multicycle case, the multicycle case by masking out edges immediately adjacent to node x in the base case.

8. The method of claim 6 wherein cut-path constraints on timing analysis are treated as multicycle constraints with an infinite multicycle period.

9. The method of claim 4 wherein the resulting timing analysis is used in combination with a placement algorithm to effect a modified placement of critical edges.

10. The method of claim 4 wherein generating the edge mask further comprises generating source-edge-masks by depth-first search from destination registers or pins in which edge-masks on a node or edge are defined as the inclusive OR of the edge masks of their fanin edges or nodes.

11. The method of claim 4 wherein generating the edge masks further comprises generating destination-edge-masks by depth-first search from source registers or pins in which edge-masks on a node or edge are defined as the inclusive OR of the edge masks of their fanout edges or nodes.

12. The method of claim 4 wherein generating the edge masks further comprises generating one or more super-edge masks that are used to represent multiple constraint types merged into a single constraint.

13. The method of claim 9 wherein generating the edge masks further comprises generating additional edge masks to represent critical and non-critical portions of a netlist for the user design, and the method further comprises:

applying timing analysis on the critical portion of the netlist with greater frequency; and

applying timing analysis on the non-critical portion of the netlist less frequently.

14. The method of claim 9 wherein the edge masks are used to identify constraint domains in which placement changes are made, counters are made of the number

of constraint domains, and placement requests timing analysis only on the constraint domains that have changed as the result of placement.

15. The method of claim 3 further comprising:

annotating the graph with a number of paths that pass each of the edges; and

successively pruning each edge, by setting the edge masks, until only k paths remain to be reported to a user of a timing analysis tool.

16. The method of claim 15 wherein information reported to the user is a set of k most critical paths within each constraint domain, reported separately.

17. The method of claim 16 wherein the number, k, is different for each constraint domain.

18. A computer system for implementing timing analysis on a user design that has been placed on a programmable integrated circuit, the computer system comprising:

code for generating edge masks to annotate edges in a graph that represents at least a portion of the user design, the edge masks indicating whether a source point and a destination point are reachable from a corresponding one of the edges;

code for performing at least one depth first search along a time critical path in the graph between the source point and the destination point, the at least one depth first search being prevented by the edge masks from analyzing paths that do not connect the source and the destination points; and

a computer readable medium that stores the codes.

19. The computer system according to claim 18 further comprising:

code for calculating slack and slack ratio values for edges in the time critical paths; and

code for modifying placement of the user design within the programmable integrated circuit using the slack and the slack ratio values, if delay along any of the time critical paths exceeds the user timing constraint.

20. The computer system according to claim 19 wherein the code for generating the edges masks further comprises code for generating a first type of edge mask that represents reachabiltiy from the source points in a backwards direction, and a second type of edge mask that represents reachability to the destination points in a forward direction.

21. The computer system according to claim 20 wherein the code for generating the edge masks further comprises code for generating binary bits that correspond to specific types of the source points and the destination points in a list of constraints.

22. The computer system according to claim 21 wherein the code for performing the at least one depth first search further comprises code for performing multiple depths first searches in the presence of multiple constraints on timing analysis, wherein source edge-masks correspond to source-types in timing constraints.

23. The computer system according to claim 22 further comprising:

code for converting multicycle constraints on timing analysis to two edge-mask sets, one for a base case and another for a multicycle case.

24. The computer system according to claim 22 further comprising:

code for converting thru-x constraints into multiple edge-mask sets indicating a base case and a multicycle case, the multicycle case by masking out edges immediately adjacent to node x in the base case.

25. The computer system according to claim 23 wherein cut-path constraints on timing analysis are treated as multicycle constraints with an infinite multicycle period.

26. The computer system according to claim 21 wherein the code for generating the edge mask further comprises code for generating source-edge-masks by depth-first search from destination registers or pins in which edge-masks on a node or edge are defined as the inclusive OR of the edge masks of their fanin edges or nodes.

27. The computer system according to claim 21 wherein the code for generating the edge mask further comprises code for generating destination-edge-masks by depth-first search from source registers or pins in which edge-masks on a node or edge are defined as the inclusive OR of the edge masks of their fanout edges or nodes.

28. The computer system according to claim 21 wherein the code for generating the edge masks further comprises code for generating one or more super-edge masks that are used to represent multiple constraint types merged into a single constraint.

29. The computer system according to claim 21 wherein the code for generating the edge masks further comprises code for generating edge masks to represent critical and non-critical portions of a netlist for the user design, and the computer system further comprises:

code for applying timing analysis on the critical portion of the netlist with greater frequency; and

code for applying timing analysis on the non-critical portion of the netlist less frequently.

30. The computer system according to claim 21 wherein the edge masks are used to identify constraint domains in which placement changes are made, counters are made of the number of constraint domains, and placement requests timing analysis only on the constraint domains that have changed as the result of placement.

31. The computer system according to claim 20 further comprising:

code for annotating the graph with a number of paths that pass each of the edges; and

code for successively pruning each edge, by setting the edge masks, until only k paths remain to be reported to a user of a timing analysis tool.